A collection of papers on the fundamentals of software analysis with a focus on concurrency, race detection, and partial ordering techniques. (Original List from Paper list: Fundamentals of Software Analysis, Jeff Huang, Spring 2018, https://parasol.tamu.edu/~jeff/course/689_spring2018/papers.html)

Partial Ordering

- *(DC analysis + Vindicator)* High-Coverage, Unbounded Sound Predictive Race Detection. Jake Roemer, Kaan Genç, Michael D. Bond. (PLDI 2018)
- *(Schedulable Happens-Before)* What Happens-After the First Race? Enhancing the Predictive Power of Happens-Before Based Dynamic Race Detection. Dileep Kini, Umang Mathur, Mahesh Viswanathan. (OOPSLA 2018)
- *(Weak Causally Precedes)* Dynamic Race Prediction in Linear Time. Dileep Kini, Umang Mathur, Mahesh Viswanathan. (PLDI 2017)
- *(CP-sub polynomial time)* An Online Dynamic Analysis for Sound Predictive Data Race Detection. Jake Roemer, Michael D. Bond. (Technical Report OSU-CISRC-11/16-TR05, 2016)
- *(Maximal Causality)* Stateless Model Checking Concurrent Programs with Maximal Causality Reduction. Jeff Huang. (PLDI 2015)
- *(Causally Precedes)* Sound Predictive Race Detection in Polynomial Time. Yannis Smaragdakis, Jacob M. Evans, Caitlin Sadowski, Jaeheon Yi, Cormac Flanagan. (POPL 2012)
- *(Lockset discipline)* Eraser: A Dynamic Data Race Detector for Multithreaded Programs. Stefan Savage et al. (TOCS 1997)
- *(Happens-Before)* Time, Clocks, and the Ordering of Events in a Distributed System. Leslie Lamport. (1978)

Static Analysis

- FlowDroid: Precise Context, Flow, Field, Object-sensitive and Lifecycle-aware Taint Analysis for Android Apps. S. Arzt et al. (PLDI 2014)
- Pick Your Contexts Well: Understanding Object-Sensitivity. Yannis Smaragdakis, Martin Bravenboer, and Ondrej Lhotak. (POPL 2011)
- A few Billion Lines of code Later using static Analysis to find Bugs in the Real World. Dawson Engler et al. (CACM 2010)
- Effective Static Race Detection for Java. Mayur Naik, Alex Aiken, and John Whaley. (PLDI 2006)
- Cloning-based context-sensitive pointer alias analysis using binary decision diagrams. John Whaley and Monica S. Lam. (PLDI 2004)
- A Type and Effect System for Atomicity. Cormac Flanagan and Shaz Qadeer. (PLDI 2003)
- A Static Analyzer for Large Safety-Critical Software. Bruno Blanchet et al. (PLDI 2003)

- Scalable Propagation-Based Call Graph Construction Algorithms. Frank Tip and Jens Palsberg. (OOPSLA 2000)
- Type-Based Race Detection for Java. Cormac Flanagan and Stephen N. Freund. (PLDI 2000)
- Program Analysis via Graph Reachability. Thomas Reps. (ISLP 1997)

Dynamic Analysis

- EffectiveSan: Type and Memory Error Detection Using Dynamically Typed C/C++. Gregory J. Duck, Roland H. C. Yap. (PLDI 2018)
- AddressSanitizer: A Fast Address Sanity Checker. Konstantin Serebryany, Derek Bruening, Alexander Potapenko, Dmitry Vyukov. (USENIX ATC 2012)
- All You Ever Wanted to Know About Dynamic Taint Analysis and Forward Symbolic Execution. Edward J. Schwartz, Thanassis Avgerinos, and David Brumley. (OAKLAND 2010)
- ThreadSanitizer -- data race detection in practice. Konstantin Serebryany, Timur Iskhodzhanov. (WBIA 2009)
- How to Shadow Every Byte of Memory Used by a Program. N. Nethercote and J. Seward (VEE 2007)
- Whole Execution Traces. X. Zhang and R. Gupta. (MICRO 2004)
- Static and Dynamic Analysis: Synergy and Duality. Michael D. Ernst. (WODA 2003)
- Precise Dynamic Slicing Algorithms. X. Zhang and R. Gupta. (ICSE 2003)
- Dynamically Discovering Likely Program Invariants to Support Program Evolution. M. D. Ernst, J. Cockrell, W. G. Griswold, and D. Notkin. (TSE 2001)
- Whole Program Paths. J. Larus. (PLDI 1999)
- Efficient Path Profiling. T. Ball and J. Larus. (MICRO 1996)

Symbolic Execution and Testing

- A Survey of Symbolic Execution Techniques. Roberto Baldoni, Emilio Coppa, Daniele Cono DÕElia, Camil Demetrescu, and Irene Finocchi. (arXiv last update: 2017)
- Enhancing Symbolic Execution with Veritesting. Thanassis Avgerinos, Alexandre Rebert, Sang Kil Cha, and David Brumley. (ICSE 2014)
- Jalangi: A Selective Record-Replay and Dynamic Analysis Framework for JavaScript. K. Sen, S. Kalasapur, T. Brutch, and S. Gibbs (FSE 2013)
- Symbolic PathFinder: Symbolic Execution of Java Bytecode. C. S. Pasareanu and N. Rungta. (ASE 2010)
- Execution Synthesis: A Technique for Automated Software Debugging. Cristian Zamfir and George Candea. (EUROSYS 2010)
- KLEE: Unassisted and Automatic Generation of High-Coverage Tests for Complex Systems Programs. C. Cadar, D. Dunbar, and D. Engler. (OSDI 2008)
- DART: Directed Automated Random Testing. Patrice Godefroid, Nils Klarlund, and Koushik Sen. (PLDI 2005)

Debugging and Bug Finding

- Compiler Validation via Equivalence Modulo Inputs. Vu Le, Mehrdad Afshari, and Zhendong Su. (PLDI 2014)
- Precise Memory Leak Detection for Java Software using Container Profiling. G. Xu and A. Rountev. (ICSE 2008)
- Scalable Statistical Bug Isolation. B. Liblit, M. Naik, A. X. Zheng, A. Aiken, and M. I. Jordan. (PLDI 2005)
- Finding Bugs is Easy. David Hovemeyer and William Pugh. (SIGPLAN NOTICES 2004)
- CP-Miner: A Tool for Finding Copy-paste and Related Bugs in Operating System Code. Z. Li, S. Lu, S. Myagmar, and Y. Y. Zhou. (OSDI 2004)
- Simplifying and Isolating Failure-Inducing Input. A. Zeller and R. Hildebrandt. (TSE 2002)
- The SLAM Project: Debugging System Software via Static Analysis. Thomas Ball and Sriram K. Rajamani. (POPL 2002)

Security

- Meltdown. Moritz Lipp et al. (2018)
- Spectre Attacks: Exploiting Speculative Execution. Paul Kocher et al. (2018)
- Cimplifier: Automatically Debloating Containers. Vaibhav Rastogi, Drew Davidson, Lorenzo De Carli, Somesh Jha, Patrick McDaniel (FSE 2017)
- Code-Pointer Integrity. Volodymyr Kuznetsov, Laszl o Szekeres, Mathias Payer, George Candea, R. Sekar, Dawn Song (OSDI 2014)
- Automatic exploit generation. T Avgerinos, SK Cha, A Rebert, EJ Schwartz, M Woo, D Brumley (CACM 2014)
- Control-Flow Integrity Principles, Implementations, and Applications. Martin Abadi, Mihai Budiu, òlfar Erlingsson, Jay Ligatti (TISSEC 2009)
- Preventing memory error exploits with WIT. Periklis Akritidis, Cristian Cadar, Costin Raiciu, Manuel Costa, Miguel Castro (IEEE S&P 2008)
- Securing software by enforcing data-flow integrity. Miguel Castro, Manuel Costa, Tim Harris (OSDI 2006)
- Remote timing attacks are practical. David Brumley, Dan Boneh (USENIX SECURITY 2003)

Program Analysis Frameworks

- Angr -- The Next Generation of Binary Analysis. Fish Wang, Yan Shoshitaishvili (IEEE S&P 2016)
- BAP: A binary analysis platform. David Brumley, Ivan Jager, Thanassis Avgerinos, Edward J Schwartz (CAV 2011)
- Soot: The Soot framework for Java program analysis: a retrospective. Patrick Lam, Eric Bodden, Ondrej Lhotak, and Laurie Hendren (CETUS 2011)

- WALA: Static and Dynamic Program Analysis using WALA. Julian Dolby and Manu Sridharan. (PLDI Tutorial 2010)
- RoadRunner: The RoadRunner Dynamic Analysis Framework for Concurrent Programs. Cormac Flanagan and Stephen N. Freund. (PASTE 2010)
- CalFuzzer: An Extensible Active Testing Framework for Concurrent Programs. P. Joshi, M. Naik, C.-S. Park, and K. Sen. (CAV 2009)
- Valgrind: A Framework for Heavyweight Dynamic Binary Instrumentation. N. Nethercote and J. Seward. (PLDI 2007)
- Pin: Building Customized Program Analysis Tools with Dynamic Instrumentation. C. K. Luk et al. (PLDI 2005)
- Jikes RVM: The Jikes Research Virtual Machine project: Building an open-source research community. B. Alpern et al. (IBM Systems Journal 2005)
- LLVM: A Compilation Framework for Lifelong Program Analysis & Transformation . Chris Lattner and Vikram Adve. (CGO 2004)
- Java PathFinder: Test Input Generation with Java PathFinder. W. Visser, C. S. Pasareanu, and S. Khurshid. (ISSTA 2004)

Concurrency in Practice

- Java concurrency bug patterns for multicore systems. Zhi Da Luo, Yarden Nir-Buchbinder, and Raja Das. (IBM 2010)
- Learning from Mistakes - A Comprehensive Study on Real World Concurrency Bug Characteristics. S. Lu, S. Park, E. S. and Y. Y. Zhou. (ASPLOS 2008)
- Java Concurrency in Practice. Brian Goetz, Tim Peierls, Joshua Bloch, Joseph Bowbeer, David Holmes, and Doug Lea. (BOOK 2006)
- Concurrent Bug Patterns and How to Test Them. Eitan Farchi, Yarden Nir, Shmuel Ur. (IPDPS 2003)

Data Races

- RacerD: Compositional Static Race Detection. Sam Blackshear, Nikos Gorogiannis, Peter W. O'Hearn, Ilya Sergey. (OOPLSA 2018)
- ECHO: Instantaneous In Situ Race Detection in the IDE. Sheng Zhan, Jeff Huang. (FSE 2016)
- What's the Optimal Performance of Precise Dynamic Race Detection?--A Redundancy Perspective. Jeff Huang, Arun K. Rajagopalan. (ECOOP 2015)
- Commutativity Race Detection. Dimitar Dimitrov, Veselin Raychev, Martin Vechev, and Eric Koskinen. (PLDI 2014)
- Maximal Sound Predictive Race Detection With Control Flow Abstraction. Jeff Huang, Patrick Meredith, and Grigore Rosu. (PLDI 2014)
- Practical Static Race Detection for Java Parallel Loops. Cosmin Radoi, Danny Dig. (ISSTA 2013)
- Data Races vs. Data Race Bugs: Telling the Difference with Portend. Baris Kasikci, Cristian ZamÞr, and George Candea. (ASPLOS 2012)
- Detecting and Surviving Data Races using Complementary Schedules. K.

Veeraraghavan, P. M. Chen, J. Flinn, S. Narayanasamy. (SOSP 2011)
- Locksmith: Practical Static Race Detection for C. Polyvios Pratikakis, Jeffrey S. Foster, Michael Hicks. (ACM Transactions on Programming Languages and Systems 2011)
- Effective Data-Race Detection for the Kernel. John Erickson, Madanlal Musuvathi, Sebastian Burckhardt, Kirk Olynyk. (OSDI 2010)
- Pacer: Proportional Detection of Data Races. Michael D. Bond, Katherine E. Coons, Kathryn S. McKinley. (PLDI 2010)
- FastTrack: Efficient and Precise Dynamic Race Detection. Cormac Flanagan, Stephen N. Freund. (PLDI 2009)
- LiteRace: Effective Sampling for Lightweight Data-Race Detection. Daniel Marino, Madanlal Musuvathi, Satish Narayanasamy. (PLDI 2009)
- Race Directed Random Testing of Concurrent Programs. Koushik Sen. (PLDI 2008)
- RELAY: Static Race Detection on Millions of Lines of Code. Jan Wen Voung, Ranjit Jhala, Sorin Lerner. (FSE 2007)
- Goldilocks: A Race and Transaction-Aware Java Runtime. Tayfun Elmas, Shaz Qadeer, Serdar Tasiran. (PLDI 2007)
- RacerX: effective, static detection of race conditions and deadlocks. Dawson Engler, Ken Ashcraft. (SOSP 2003)
- Hybrid Dynamic Data Race Detection. Robert O'Callahan and Jong-Deok Choi. (PPOPP 2003)
- Efficient on-the-fly data race detection in multithreaded C++ programs. Eli Pozniansky, Technion, Assaf Schuster. (PPoPP 2003)
- Eraser: A Dynamic Data Race Detector for Multithreaded Programs. Stefan Savage et al. (TOCS 1997)
- What are Race Conditions: Some Issues and Formalizations. Robert Netzer and Barton Miller. (LOPLAS 1992)

Atomicity, Serializability, and Linearizability

- Detecting Atomic-Set Serializability Violations in Multithreaded Programs through Active Randomized Testing. Zhifeng Lai et al. (ICSE 2010)
- Line-Up: A Complete and Automatic Linearizability Checker. Sebastian Burckhardt, Chris Dern, Madanlal Musuvathi, Roy Tan. (PLDI 2010)
- CTrigger: Exposing Atomicity Violation Bugs from Their Hiding Places. Soyeon Park, Shan Lu, and Yuanyuan Zhou. (ASPLOS 2009)
- Velodrome: A Sound and Complete Dynamic Atomicity Checker for Multithreaded Programs. C. Flanagan, S. N. Freund, J. Yi. (PLDI 2008)
- Atomizer: A Dynamic Atomicity Checker for Multithreaded Programs. Cormac Flanagan, Stephen N. Freund. (POPL 2004)

Deadlocks

- ConLock: A Constraint-Based Approach to Dynamic Checking on Deadlocks in

Multithreaded Programs. Yan Cai et al. (ICSE 2014)
- A Randomized Dynamic Program Analysis Technique for Detecting Real Deadlocks. Pallavi Joshi, Chang-Seo Park, Koushik Sen, Mayur Naik. (PLDI 2009)
- Effective Static Deadlock Detection. Mayur Naik, Chang-Seo Park, Koushik Sen, and David Gay. (ICSE 2009)
- Gadara: Dynamic Deadlock Avoidance for Multithreaded Programs. Yin Wang et al. (OSDI 2008)

Partial Order Reduction

- Stateless Model Checking Concurrent Programs with Maximal Causality Reduction. Jeff Huang. (PLDI 2015)
- Dynamic partial-order reduction for model checking software. C. Flanagan and P. Godefroid. (POPL 2005)
- State Space Reduction using Partial Order Techniques. E.M. Clarke, O. Grumberg, M. Minea, D. Peled. (STTT 1998)
- Partial-Order Methods for the Verification of Concurrent Systems: An Approach to the State-Explosion Problem. P Godefroid. (PhD thesis, University of LiÃ´lge., 1996)

Testing, Isolation, and Repairing

- Maple: A Coverage-Driven Testing Tool for Multithreaded Programs. Jie Yu, Satish Narayanasamy, Cristiano Pereira, and Gilles Pokam. (OOPSLA 2012)
- Automated Concurrency-Bug Fixing. Guoliang Jin, Wei Zhang, Dongdong Deng, Ben Liblit, and Shan Lu. (OSDI 2012)
- AXIS: Automatically Fixing Atomicity Violations Through Solving Control Constraints. Peng Liu and Charles Zhang. (ICSE 2012)
- Laws of Order: Expensive Synchronization in Concurrent Algorithms Cannot be Eliminated. Hagit Attiya et al. (POPL 2011)
- ISOLATOR: Dynamically Ensuring Isolation in Concurrent Programs. Sriram Rajamani et al. (ASPLOS 2009)
- Finding and Reproducing Heisenbugs in Concurrent Programs. Madanlal Musuvathi et al. (OSDI 2008)
- Iterative Context Bounding for Systematic Testing of Multithreaded Programs. Madan Musuvathi and Shaz Qadeer. (PLDI 2007)

Memory Consistency Models

- End-To-End Sequential Consistency. Abhayendra Singh, Satish Narayanasamy, Daniel Marino, Todd Millstein, and Madanlal Musuvathi. (ISCA 2012)
- A Case for an SC-Preserving Compiler. Daniel Marino, Abhayendra Singh, Todd Millstein, Madanlal Musuvathi, and Satish Narayanasamy. (PLDI 2011)
- A Primer on Memory Consistency and Cache Coherence. Daniel J. Sorin, Mark D. Hill, and David A. Wood. (BOOK 2011)

- Memory Models: A Case for Rethinking Parallel Languages and Hardware. Sarita V. Adve, Hans-J. Boehm. (CACM 2010)
- Adversarial Memory For Detecting Destructive Races. Cormac Flanagan, Stephen N. Freund. (PLDI 2010)
- DRFx: A Simple and Efficient Memory Model for Concurrent Programming Languages. Daniel Marino et al. (PLDI 2010)
- MemSAT: Checking Axiomatic Speciþcations of Memory Models. Emina Torlak, Mandana Vaziri, and Julian Dolby. (PLDI 2010)
- The Java Memory Model. Jeremy Manson, William Pugh, Sarita V. Adve. (POPL 2005)

Multithreaded Record and Replay

- CLAP: Recording Local Executions to Reproduce Concurrency Failures. Jeff Huang, Charles Zhang, and Julian Dolby. (PLDI 2013)
- DDOS: Taming Nondeterminism in Distributed Systems. Nicholas Hunt, Tom Bergan, Luis Ceze, Steven D. Gribble. (ASPLOS 2013)
- DoublePlay: Parallelizing Sequential Logging and Replay. Kaushik Veeraraghavan et al. (ASPLOS 2011)
- LEAP: Lightweight Deterministic Multi-processor Replay of Concurrent Java Programs. Jeff Huang, Peng Liu, and Charles Zhang. (FSE 2010)
- PinPlay: A Framework for Deterministic Replay and Reproducible Analysis of Parallel Programs. Harish Patil et al. (CGO 2010)
- PRES: Probabilistic Replay with Execution Sketching on Multiprocessors. Soyeon Park et al. (SOSP 2009)

Deterministic Multithreading

- Dthreads: Efficient Deterministic Multithreading. Tongping Liu, Charlie Curtsinger, Emery D. Berger. (SOSP 2011)
- PARROT: A Practical Runtime for Deterministic, Stable, and Reliable Threads. Heming Cui et al. (ASPLOS 2010)
- CoreDet: A Compiler and Runtime System for Deterministic Multithreaded Execution. Tom Bergan et al. (ASPLOS 2010)
- Kendo: Efficient Deterministic Multithreading in Software. Marek Olszewski, Jason Ansel, Saman Amarasinghe. (ASPLOS 2009)
- DMP: Deterministic Shared Memory Multiprocessing. Joseph Devietti, Brandon Lucia, Luis Ceze, Mark Oskin. (ASPLOS 2009)

Concurrency Programming Models

- Grace: Safe Multithreaded Programming for C/C++. Emery D. Berger, Ting Yang, Tongping Liu, Gene Novark. (OOPSLA 2009)
- Parallel Programming Must Be Deterministic by Default. Robert L. Bocchino Jr., Vikram S. Adve, Sarita V. Adve, Marc Snir. (HotPar 2009)
- Associating Synchronization Constraints with Data in an Object-Oriented

Language. Mandana Vaziri, Frank Tip, and Julian Dolby. (POPL 2006)

Transactional Memory

- Using Hardware Memory Protection to Build a High-Performance, Strongly Atomic Hybrid Transactional Memory. L. Baugh et al. (ISCA 2008)
- Enforcing Isolation and Ordering in STM. Tatiana Shpeisman et al. (PLDI 2007)
- LogTM: Log-based Transactional Memory. Kevin E. Moore, Jayaram Bobba, Michelle J. Moravan, Mark D. Hill, and David A. Wood. (HPCA 2006)
- Hybrid Transactional Memory. Peter Damron, Alexandra Fedorova, Yossi Lev, Victor Luchangco, Mark Moir, and Daniel Nussbaum. (ASPLOS 2006)